ФПМИ БГУ

Вычислительные методы алгебры
Лаборатоная работа 5

Решение СЛАУ итерационными методами

(метод простых итераций, метод Гаусса-Зейделя)

Подготовил:
Ткачук Павел
2 курс 1 группа

Преподаватель:
Будник Анатолий Михайлович

21 декабря 2016 г.

1 Постановка задачи

Система:

$$\begin{cases} a_{1,1}x_1 + a_{1,2}x_2 + \ldots + a_{1,n}x_n = f_1 \\ a_{2,1}x_1 + a_{2,2}x_2 + \ldots + a_{2,n}x_n = f_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \\ a_{n,1}x_1 + a_{n,2}x_2 + \ldots + a_{n,n}x_n = f_n \end{cases} \tag{1}$$

Входные данные:

$$\left[\begin{array}{ccccc|c} 0.6444 & 0.0000 & -0.1683 & 0.1184 & 0.1973 & 1.2677 \\ -0.0395 & 0.4208 & 0.0000 & -0.0802 & 0.0263 & 1.6819 \\ 0.0132 & -0.1184 & 0.7627 & 0.0145 & 0.0460 & -2.3657 \\ 0.0395 & 0.0000 & -0.0960 & 0.7627 & 0.0000 & -6.5369 \\ 0.0263 & -0.0395 & 0.1907 & -0.0158 & 0.5523 & 2.8351 \end{array}\right]$$

Задача:

1. Найти решение СЛАУ x итерационными методами (метод простых итерации, метод Гаусса-Зейделя) с точностью $\varepsilon = 10^{-5}$
2. Найти вектор невязки $r = Ax - f$ и количество шагов итерационнго метода

2 Алгоритм

2.1 Метод простых итераций

1. Преобразуем систему $Ax = f$ к виду $x = Bx + g$, где
$$B = E - \frac{A^T A,}{\|A^T A\|},\ g = \frac{A^T f}{\|A^T A\|}$$
.
2. Полагаем $k = 0$, $x^{(0)} = g$ и $\varepsilon = 10^{-5}$
3. Вычисляем следующее приближение $x^{(k+1)} = Bx^{(k)} + g$
4. Если выполнено условие $\|x^{(k+1)} - x^{(k)}\| < \epsilon$, завершаем процесс и в качестве приближенного решения задачи берем $x^* \cong x^{(k+1)}$. Иначе полагаем $k = k + 1$ и переходим к пункту 3 алгоритма.

2.2 Метод Зейделя

1. Полагаем $k = 0$, $x^{(0)} = g$ и $\varepsilon = 10^{-5}$
2. Вычисляем следующее приближение
$$x_i^{(k+1)} = \frac{b_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(k)}}{a_{ii}},\ i = 1, \ldots, n.$$
3. Если выполнено условие $\|x^{(k+1)} - x^{(k)}\| < \epsilon$, завершаем процесс и в качестве приближенного решения задачи берем $x^* \cong x^{(k+1)}$. Иначе полагаем $k = k + 1$ и переходим к пункту 2 алгоритма.

3 Результаты и вывод

3.1 Входные данные

0.6444 0.0000 -0.1683 0.1184 0.1973 1.2677
-0.0395 0.4208 0.0000 -0.0802 0.0263 1.6819
0.0132 -0.1184 0.7627 0.0145 0.0460 -2.3657
0.0395 0.0000 -0.0960 0.7627 0.0000 -6.5369
0.0263 -0.0395 0.1907 -0.0158 0.5523 2.8351

3.2 Выходные данные

```
Решаем матричное уравнение итерационными методами
Основная матрица системы:
[[ 0.6444  0.     -0.1683  0.1184  0.1973]
 [-0.0395  0.4208  0.     -0.0802  0.0263]
 [ 0.0132 -0.1184  0.7627  0.0145  0.046 ]
 [ 0.0395  0.     -0.096   0.7627  0.    ]
 [ 0.0263 -0.0395  0.1907 -0.0158  0.5523]]
Свободные члены:  [ 1.2677  1.6819 -2.3657 -6.5369  2.8351]
Метод сходится
Метод простых итераций
Решаем с точность 0.000001
Решение  [ 0.99823504  1.99990915 -2.99974236 -9.00000603  6.0070324 ]
Получено за  69  итераций
Невязка  [  6.07606503e-06   1.69225939e-05   7.36262616e-06   9.50905528e-07
  -9.60824597e-06]
Норма невязки  2.16961661447e-05
Метод Зейделя
Решаем с точность 0.000001
Решение  [ 0.99821559  1.9998653  -2.99975966 -9.00000845  6.00705349]
Получено за  6  итераций
Невязка  [  3.31840453e-07  -1.22080552e-08   3.92101627e-08   0.00000000e+00
  -4.44089210e-16]
Норма невязки  3.34371888462e-07
```

3.3 Вывод

С помощью итерационных методов можно получить результат с любой точностью, однако количество шагов, требуемое для получения достаточно точного результата (в данном случае с точностью $\varepsilon = 10^{-5}$), значительно меньше, чем в точных методах (Гаусса и квадратного корня).

Количество шагов метода простых итераций на прямую зависит от выбора матрицы B и вектора g. То есть при другом выборе матрицы и вектора, метод может сойтись и быстрее.

Метод Гаусса-Зейделя сходится быстрее метода итераций, и количество шагов этого метода зависит только от исходной матрицы.

Итак, можно сделать вывод, что во многих задачах выгоднее использовать итерационные методы (например, Гаусса-Зейделя), так как можно получить решение с такой же погрешностью как и в точных методах (в точных методах появляется погрешность вычислений), однако за меньшее количество операций.

4 Листинг кода

```
import numpy as np
import numpy.linalg as linalg


def is_solvable(matr_A, matr_b): # Проверка сходимости
    A = np.array(matr_A)
    b = np.array(matr_b)
    size = len(A)
    E = np.eye(size)
    B = np.array(E - np.dot(A.transpose(), A) / linalg.norm(np.dot(A.transpose(), A)))
    sums = []
    for i in range(size):
        sums.append(sum(abs(B[i, j]) for j in range(size)))
    return max(sums) < 1

def simple_iteration(matr_A, matr_b, eps):
    A = np.array(matr_A)
    b = np.array(matr_b)
    size = len(b)
    E = np.eye(size)
    B = np.array(E - np.dot(A.transpose(), A) / linalg.norm(np.dot(A.transpose(), A)))
    g = np.array(np.dot(A.transpose(), b) / linalg.norm(np.dot(A.transpose(), A)))
    x = g;
    converge = False;
    count = 0
    while not converge:
        count +=1
        x_new = x.copy()
        x_new = np.dot(B, x) + g
        converge = linalg.norm(x - x_new) <= eps
        x = x_new

    return x, count

def seidel(matr_A, matr_b, eps):
    A = np.array(matr_A)
    b = np.array(matr_b)
    n = len(b)
    x = b.copy();
    count = 0
    converge = False
    while not converge:
        count += 1
        x_new = x.copy()
        for i in range(n):
            s1 = sum(A[i][j] * x_new[j] for j in range(i))
            s2 = sum(A[i][j] * x[j] for j in range(i + 1, n))
            x_new[i] = (b[i] - s1 - s2) / A[i][i]
        converge = linalg.norm(x - x_new) <= eps
        x = x_new

    return x, count


file = open("matrix", "r")  # Чтение файла
A, b = [], []
```

```
for line in file:
    A.append([float(el) for el in line.split()[:-1]])
    b.append(float(line.split().pop()))
A = np.array(A)
b = np.array(b)
print("Решаем матричное уравнение итерационными методами")
print("Основная матрица системы:")
print(A)
print("Свободные члены: ", b)
if is_solvable(A, b):
    print("Метод сходится")
    print("Метод простых итераций")
    ans = simple_iteration(A, b, 0.00001)
    print("Решаем с точность 0.000001")
    print("Решение ", ans[0], "\nПолучено за ", ans[1], " итераций")
    print("Невязка ", np.dot(A, ans[0]) - b)
    print("Норма невязки ", linalg.norm(np.dot(A, ans[0]) - b))
    print("Метод Зейделя")
    ans = seidel(A, b, 0.00001)
    print("Решаем с точность 0.000001")
    print("Решение ", ans[0], "\nПолучено за ", ans[1], " итераций")
    print("Невязка ", np.dot(A, ans[0]) - b)
    print("Норма невязки ", linalg.norm(np.dot(A, ans[0]) - b))
else:
    print("Метод расходится")
```